Spell Slinger

Functional Specification
4th Year Project

Student:	David Darigan
Student No:	C00263218
Supervisor:	Dr Joseph Kehoe
Submission Date:	08/12/2023

Table of Contents

Table of Figures

1. Introduction

The purpose of this functional specification document is to define what the "Spell Slinger" application is and its core functionalities. This document specifies the design and target audience for the game. A Context-Diagram is utilised to represent how data flows through the application, a Use Case diagram is used to display the the applications functionality of at a broad level, brief and detailed uses cases were written to deepen the reader's understanding of the use cases outlined in the Use Case diagram. Additional information about the application's Functionality, Usability, Reliability, Performance, Supportability and Security will be addressed in the FURPS+ section of this document.

2. Application Definition

Spell Slinger is an augmented reality android game using geolocation where users travel to real world locations in order to collect spells from Spell Wells that exist at those points and then use them as consumable items in battle against creatures or other players.

Spell Slinger has two fundamental goals:

- Build local communities through gaming

  By creating points of interest through Spell Wells, players would meet each other and bond over the common shared interest of the game itself.

- Gamify Healthy Living

  Users can also gain eXperience Points (XP) through walking which they can then allocate to their own battle attributes such as Health or Speed.

<u>A Common Playthrough</u>

Nicole logs in to her account. She opens the World Map and enables the location services. She cannot see anything on the World Map so she taps the ‘scan’ button. A Spell Well appears about 2 kilometres from her current location. She notes her experience points are currently at 2. She walks to the coordinates of the Spell Well, where she sees five others, some of them on their phone. She collects her Spells (2 Fireballs, 1 Cure and 1 Toxic) from the Spell Well. She opens her Spell Book to locate her new Toxic Spell and insert it into a free slot in her Spell Belt. She closes the Spell Book and opens her attributes. Her experience points are now showing 5. She invests three points into Health and 2 points into Fire Affinity. She closes her attributes and opens the World Map again, taps ‘scan’ and sees three more players appear near the Well. She asks who is playing, two reply, and they begin to exchange their experiences playing Spell Slinger.

3. Target Audience

The two target audiences for Spell Slinger is borne from it's fundamental goals:

- Social Gamers

  Social Gamers are people who use gaming as a medium to meet other people, with the aid of a common bond (the game in question), as a starting point to bond and form relationships and build communities with each other.

- Active Gamers

  Active Gamers are people who use gaming as a way to motivate and reward themselves into an active lifestyle. By rewarding players with XP for light exercise, they are motivated to exercise for further and longer.

4. Data Flow Diagram

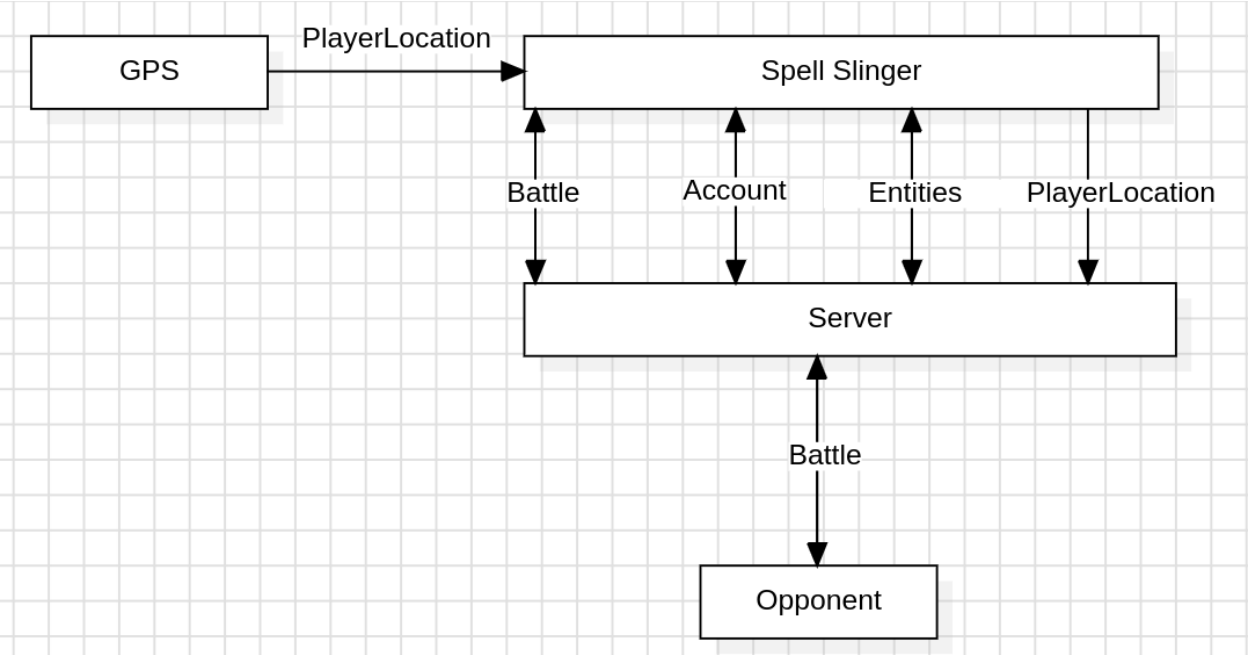

Figure 1-1 Data Flow Diagram

5. Use Case Diagram

Figure 1.2 Use Case Diagram

6. Brief Use Cases

Name	Register
Actors	Player
Description	This use case begins when a player wants to register a new account. 1. The player will enter a unique name, password and a valid email address in the register screen. 2. The server will send a link to the email address the user input with a link to verify that address. 3. The user clicks on the link in the email to verify their email address is valid. 4. This use case ends when the server registers all of the input user details into the database.

Name	Login
Actors	Player
Description	This use case begins when a player wants to login to their account. 1. The player will enter their username and password. 2. The server will confirm that the account exists with those details. 3. This use case ends when the player has successfully logged into their account.

Name	Logout
Actors	Player
Description	This use case begins when a player wants to log out of their account. 1. The user taps the ‘logout’ option. 2. This use case ends when the player has successfully logged out of their account.

Name	Delete Account
Actors	Player
Description	This use case begins when a player wants to delete their account. 1. The user opens their settings and taps the ‘Delete Account’ option. 2. The game prompts the user to enter their login credentials. 3. The game prompts the player to confirm 4. The player taps ‘confirm’. 5. This use case ends when the player’s account is deleted from the server.

Name	Train
Actors	Player, GPS
Description	This use case begins when the player has granted location services permission to the game. 1. The game periodically pings the GPS to request the player's current location. 2. The game calculates the difference in metres walked from the previous GPS location. 3. The game forwards each kilometre walked to the server. 4. The server checks the timestamp and confirms that the distance travelled is within reason for walking. 5. The server stores the kilometres travelled as Experience Points. 6. This use case ends when the player has stopped travelling.

Name	Scan
Actors	Player, GPS
Description	This use case begins when the Player taps the 'scan' button on the world map. 1. The game will ping the GPS to request the player's current location. 2. The game forwards the location to the server. 3. This use case ends when the server responds with a list of entities (wells, creatures, and or other players) near that location.

Name	Draw Spells
Actors	Player
Description	This use case begins when a player has travelled to a Spell Well. 1. The player taps on the Spell Well. 2. The Spell Well screen opens. 3. The player taps on the Spell Well on the new screen. 4. The game forwards the Spell Well location to the server. 5. This use case ends when the server adds a variable number of Spells to the player's account.

Name	Equip Spell
Actors	Player
Description	This use case begins when the player taps on their Spell Book. 1. The Spell Book screen opens. 2. The player taps on one of their Spells.

	3. The player taps on one of six slots on their Spell Belt. This use case ends when the tapped Spell is inserted into the tapped Spell Slot.

Name	Increase Stat
Actors	Player
Description	This use case begins when the player opens their attribute screen. 1. The player sets a number of points for one or more attributes (up to a total of their currently available Experience Points). 2. The player taps confirm. 3. The game prompts the player that they are sure. 4. The player taps confirm again. 5. This use case ends when the attributes and experience points are updated on the server.

Name	Battle Player
Actors	Player, Opponent
Description	This use case begins when the player taps on an Opponent. 1. A number of possible actions pop up. 2. The player taps on “challenge”. 3. A Challenge Request pop ups on the opponent’s game. 4. The opponent taps ‘accept’. 5. The battle screen opens on both games. 6. The player and the opponent take turns tapping one of their Spells to send a cast request to the server. 7. This use case ends when one player’s health has reduced to 0.

Name	Battle Creature
Actors	Player
Description	This use case begins when the user taps on a Creature. 1. The battle screen opens. 2. The player and creature take turns casting spells against each other. 3. This use case ends when either the player’s or the creature’s health is reduced to 0.

Name	View Leaderboard
Actors	Player
Description	This use case starts when the user opens the Leaderboard screen. 1. The game requests the new leaderboard rankings from the server. 2. This use case ends when the leaderboard screen is populated with new ranking data.

7. Detailed Use Cases

Name	Register
Actors	Player
Preconditions	● The “register” form is displayed ● The player is not already registered
Activity	1. The player enters a unique username 2. The player enters a password 3. The player enters a valid email address 4. The player presses “register” 5. The password is confirmed to meet security measures 6. The database confirms that the username is unique 7. The database emails a verification link 8. The screen changes to inform the user of the email 9. The player clicks on the verification link in the email 10. The email is verified 11. The account is created in the database 12. The database sends a confirmation email to the player
Postconditions	● The player account has been created successfully
Alternatives	

Name	Login
Actors	Player
Preconditions	• The player has an account • The login screen is displayed
Activity	1. The player enters their username 2. The player enters their password 3. The player selects login 4. The database confirms the login details 5. The player is logged into their account
Postconditions	• The player is logged into their account • The main screen is displayed
Alternatives	

Name	Logout
Actors	Player
Preconditions	• The player is logged in
Activity	1. The player taps the logout button 2. The player checks to make sure the player is logged in 3. The server records the player's current XP 4. The server logs the player out
Postconditions	• The player is logged out
Alternatives	

Name	Delete Account
Actors	Player
Preconditions	• The player is logged in
Activity	1. The player opens Settings 2. The player taps ‘Delete Account’ 3. The game requests the users login credentials 4. The player enters their login credentials 5. The game requests confirmation from the player 6. The player taps ‘confirm’ 7. The server deletes the Player’s account
Postconditions	• The player’s account is deleted • The player is redirected to the logged out home screen
Alternatives	

Name	Train
Actors	Player, GPS API
Preconditions	• The player has enabled Android’s ‘Location Services’ • The player has granted location permissions to the application
Activity	1. The player travels 2. The game gets the current coordinates from the GPS 3. The game calculates the distance between the current coordinates and the previous coordinates as metres 4. The game sends every kilometre travelled to the database 5. The server compares the kilometre with a timestamp of the previous kilometres travelled to make sure it is within a reasonable limit 6. The serve stores the kilometres as experience points in the Player’s account.
Postconditions	• The player has gained experience points
Alternatives	1. 5B.1 The player is moving too fast to be eligible for XP gain

Name	Scan
Actors	Player, GPS
Preconditions	● The user has enabled Android's 'Location Services' ● The user has granted location permissions to the application ● The world map is displayed
Activity	1. The player taps the 'scan' button 2. The game requests the players current location from the GPS 3. The game forwards the location to the server 4. The server returns a list of entities (wells, creatures, players) around that location 5. The World Map is populated from the entities returned by the server
Postconditions	● The World Map is populated with entities
Alternatives	

Name	Draw Spell
Actors	Player
Preconditions	• The user has located a Spell Well
Activity	1. The user taps on the Spell Well icon on the world map 2. The Spell Well screen opens 3. The user taps on the Spell Well on the new screen 4. The server generates random Spells 5. The generated Spells are added to the users account 6. The time the player used the Spell Well is used is saved in their account
Postconditions	• The player has gained spells • The player cannot use that well again for a short time period
Alternatives	

Name	Equip Spell
Actors	User
Preconditions	● The player has at least one Spell ● The SpellBook is open
Activity	1. The player taps on a Spell in their Spell Book 2. The player taps on one of their six-slots in their Spell Belt 3. The tapped Spell is inserted into the tapped Spell Belt Slot.
Consequences	● The player has at least one Spell equipped
Alternatives	

Name	Increase Stat
Actors	Player
Preconditions	● The user has gained XP (Experience Points) ● The ‘Attribute’ screen is open
Activity	1. The player select an attribute 2. The player sets a range of XP using a slider 3. The player selects ‘confirm’ 4. The game prompts the player to confirm again 5. The player selects ‘confirm’ again 6. The server reduces the XP spent from the player’s account 7. The server updates the selected attribute in the player’s account
Consequences	● The player’s XP is reduced ● The selected attribute is updated
Alternatives	

Name	Battle Player
Actors	Player, Opponent
Preconditions	● The user has located an opponent

Activity	1. The player selects the opponent in the world map
	2. The player is presented with an options menu
	3. The player selects the “Challenge” option
	4. The challenge is presented to the opponent
	5. The opponent accepts the challenge
	6. The battle screen opens
	7. While player.health > 0 & opponent health > 0
	  a. The player casts a spell from their Spellbelt
	  b. The server checks that the play is valid
	  c. The opponent casts a spell from their Spellbelt
	  d. The server checks that the play is valid
	  e. The spell effects are applied by the server
	8. The player wins

Consequences	● The player’s victory is recorded by the server ● The player’s spell inventory is reduced ● The player’s rankings has increased ● The player returns to the world map
Alternatives	1. 4B. The opponent challenges the player and the player accepts 2. 8B. The opponent wins and the consequences are therefore changed.

Name	Battle Creature
Actors	Player
Preconditions	● The user has located a creature on the world map
Activity	1. The user taps on the creature in the world map 2. The battle screen opens 3. While creatures.health > 0: a. The user selects a spell from their SpellBelt b. The spell applies its effect against the creature c. The creature responds with an attack 4. The player is returned to the world map
Consequences	● The player's victory is recorded ● The player spell's inventory changes ● The creature icon is removed from the world map
Alternatives	3B. 1 Player's health reaches 0 3B. 2 Player is returned to the world map

Name	View Leaderboard
Actors	Player
Preconditions	
Activity	1. The user opens the leaderboard screen 2. The game requests new ranking data from the server 3. The server sends the ranking data to the game
Consequences	The leaderboard is updated with the new rankings
Alternatives	

8. FURPS+

The following section describes the non-functional requirements for the Spell Slinger application.

Functionality

The previous sections of this document describe the key functionalities of the Spell Slinger application.

Usability

The user interface, accessibility and responsiveness of the application are detailed in this section.

- Players shouldn't have to navigate to more than one screen from the primary screen in order to perform any action.
- The response of any new screen loading should be under 10 seconds at most (and returning to the primary screen should be faster).
- The application should use distinguishing shapes as well as complimentary colours in order to prevent ambiguity about user interface controls.

Reliability

This section specifies the intended server uptime, the reliability of the Location Services and the reconciliation processes in place for possible data loss.

- The application will use google cloud functions instead of a dedicated server in order to perform operations on the firebase realtime database which improves uptime through distributed operations.*
- The application will record distance travelled within a regular and often interval less than every 30 minutes. If the application cannot send this data to the server, it will store it locally until it can.

Performance

This section details the efficiency of the application and the rate at which the application retrieves data.

- This application will only collect world map location data on the first initial load, and then anytime a player uses the scan option to update the surrounding world with entities, by limiting our data retrieval to a scan update, we can reduce the impact of any slow response times.

- Some data server side, such as the next spells to be drawn from a Spell Well, will be precalculated, so no time is spent calculating at the time of draw.

Supportability

This section explains the rationale behind the decision on which platforms to support.

- Spell Slinger targets Android because Android is 70% of the mobile market.

  (https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/)

- Starting August 31st 2023, all new Android applications for the google play store must target Android 13 therefore Spell Slinger will target Android 13.

  (https://support.google.com/googleplay/android-developer/answer/11926878?)

+

This section considers other factors such as security

- Security will be handled Firebase Authentication performs tasks such as registering users, logging users in as well managing two factor authentication and single-sign on through a number of integrations (such as a Google or Discord account)

9. Conclusion

In conclusion, Spell Slinger is a modern Android application that is designed to encourage gamers to get out to explore and exercise in their local communities while finding other players playing it.

The main functionalities of Spell Slinger is to allow players to draw spells from Spell Wells and then use them in battle against non-player creatures or other players.

The non-functional requirements were discussed in the “FURPS+” section for Functionality, Usability, Reliability, Performance, Supportability and Security.